计算机学院专业必修课

计算机组成

多周期处理器形式建模综合方法

高小鹏

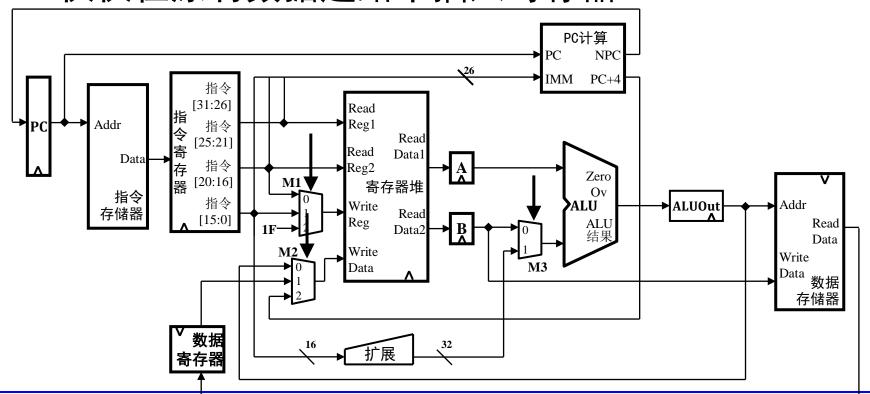
北京航空航天大学计算机学院

提纲

- 数据通路一般性方法
- ▶ 状态/控制信号矩阵压缩表达
- 2个数据通路设计的对比

数据通路的一般性方法

- 与单周期没有本质差别
- 增加:IR、A/B、ALUOut、DR
 - □ 仅仅在原有数据通路中插入寄存器



数据通路的一般性方法

- 扩展了表格
 - □ 增加IR、A/B、ALUOut、DR
- 分析方法没有任何变化
 - □ 指令执行的实质路径不因多周期设计而变化

										7						
	NPC	5	10.0	l ID		RF		•			EVT	Α	LU	ALLIQUA	DM	5
PC	IMM	PC	IM	IR	WrData	WrReg		А	В		EXT	1	2	ALUOut	DIVI	DR
						\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	V						'			

数据通路的一般性方法

- 数据通路基本特点:大部分部件是固定输入, 只有少数为多输入
 - □ 通用寄存器的目的寄存器: R、I、JAL
 - □ 通用寄存器的回写数据:运算、load、JAL
 - □ ALU的2输入

		NPC	DC	ID 4	ID		RF	Δ.	В	EVT	A	LU	ALLIQUA	DNA	DB
	PC	IMM	PC	IM	IR	WrData	WrReg	A	В	EXT	1	2	ALUOut	DM	DR
指 令						变	变					变			

示例: LW

RTL

PC-PC+4

 $R[rt] \leftarrow MEM[R[rs] + sign_ext(imm16)]$

- 主要关注: 3个
 - WrReg, WrData
 - □ ALU第2输入

		NPC	200	10.4			RF			EV.		LU			-
	PC	IMM	PC	IM	IR	WrData	WrReg	Α	В	EXT	1		ALUOut	DM	DR
LW	PC	IR[15:0]	NPC.NPC	PC	IM	DR	IR[20:16]	RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut	DM

示例: SW

SW

RTL

MEM[R[rs]+sign_ext(imm16)] \leftarrow R[rt]

PC \leftarrow PC+4

		NPC	PC	IM	IR		RF	Α	В	EXT	A	LU	ALUOut	DM		DR
	PC	IMM		IIVI		WData	RD	A		LXI	1	2	ALOGUL	Α	D	DK
LW	PC	IR[15:0]	NPC.NPC	PC	IM	DR	IR[20:16]	RF.RD1	RF.RD2	IR[15:0]	A	XT	ALU	ALUOut		DM
SW	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut	В	
									•	•		ं ना		レハナリ	기	1-Mr

示例: ADDU

ADDU

		NPC	DC.	IM	IR		RF		В	EXT	A	LU	ALLIQUE	DM		DB
	PC	IMM	PC	TIVI	IK	WData	RD	A	В	EXI	1	2	ALUOut	Α	D	DR
LW	PC	IR[15:0]	NPC.NPC	PC	IM	DR	IR[20:16]	RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut		DM
SW	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut	В	
ADDU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
						-	-	•				- 1	1215 1211 111.121	八八十 / 1	デルル	

示例: SUBU

SUBU

RTL
$$R[rd] \leftarrow R[rs] - R[rt]$$

$$PC \leftarrow PC + 4$$

		NPC	D.C.	IM	IR		RF	Δ.	В	EVT	А	LU	ALLIQuit	DM		DR
	PC	IMM	PC	IIVI	IK	WData	RD	A	В	EXT	1	2	ALUOut	А	D	DK
LW	PC	IR[15:0]	NPC.NPC	PC	IM	DR	IR[20:16]	RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut		DM
SW	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut	В	
ADDU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
SUBU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
.												ना	ユスパンツ し ユニンツ	八八十八	开ツい	1-12r

示例: ORI

ORI

		NPC	PC	IM	IR		RF	Α	В	EXT	A	LU	ALUOut	DM		DR
	PC	IMM	PC	IIVI	É	WData	RD	ţ	٥	LAI	1	2	ALOOUT	А	D	DR
LW	PC	IR[15:0]	NPC.NPC	PC	IM	DR	IR[20:16]	RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut		DM
SW	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut	В	
ADDU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
SUBU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
ORI	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[20:16]	RF.RD1		IR[15:0]	A	EXT	ALU			
			_													

示例: LUI

- 设计方案1
 - □ 延用R类、I类

RTL $R[rt] \leftarrow imm16 \mid\mid 0^{16}$ $PC \leftarrow PC+4$

		NPC	PC	IM	IR		RF	Α	В	EXT	Α	LU	ALUOut	DM		DR
	PC	IMM	PC	IIVI	IK _	WData	RD	A	В	EXI	1	2	ALOUUT	А	۵	DK
LW	PC	IR[15:0]	NPC.NPC	PC	IM	DR	IR[20:16]	RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut		DM
SW	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut	В	
ADDU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
SUBU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
ORI	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[20:16]	RF.RD1		IR[15:0]	A	EXT	ALU			
LUI	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[20:16]	RF.RD1		IR[15:0]	A	EXT	ALU			

示例: LUI

- 设计方案2
 - □减少部件

RTL $R[rt] \leftarrow imm16 \mid\mid 0^{16}$ $PC \leftarrow PC+4$

		NPC	PC	IM	IR	I	RF	Α	В	EXT	А	LU	ALUOut	DM		DR
	РС	IMM	PC	IIVI	IK	WData	RD	4	D	EXI	1	2	ALOOUL	Α	D	DK
LW	PC	IR[15:0]	NPC.NPC	PC	IM	DR	IR[20:16]	RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut		DM
SW	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut	В	
ADDU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
SUBU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
ORI	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[20:16]	RF.RD1		IR[15:0]	A	EXT	ALU			
LUI	PC	IR[15:0]	NPC.NPC	PC	IM	EXT	IR[20:16]	RF.RD1		IR[15:0]						
			_	·												

示例: BEQ

RTL
$$PC \leftarrow (R[rs]==R[rt])? PC+4+sign_ext(imm16) || 00:$$

$$PC+4$$

		NPC	PC	IM	IR		RF	Α	В	EXT	Α	LU	ALUOut	DM		DR
	PC	IMM	PC	IIVI	IK	WData	RD	A	В.	EXI	1	2	ALOUUT	А	D	DK
LW	PC	IR[15:0]	NPC.NPC	PC	IM	DR	IR[20:16]	RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut		DM
SW	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut	В	
ADDU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
SUBU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
ORI	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[20:16]	RF.RD1		IR[15:0]	A	EXT	ALU			
LUI	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[20:16]	RF.RD1		IR[15:0]	A	EXT	ALU			
BEQ	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2		A	В	ALU			
												ना	עמ בובעו עמ צוג ב		チャルし	エルルー

示例: JAL

RTL
$$R[31] \leftarrow PC + 4$$

$$PC \leftarrow PC[31:28] \parallel instr_index \parallel 00$$

ALU

DM

			PC	IM	IR			Α	В	EXT			ALUOut			DR
	PC	IMM		IIVI	IK	WData	RD	A	Б	LAI	1	2	ALOOUL	A	D	DK
LW	PC	IR[15:0]	NPC.NPC	PC	IM	DR	IR[20:16]	RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut		DM
SW	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2	IR[15:0]	A	EXT	ALU	ALUOut	В	
ADDU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
SUBU	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[15:11]	RF.RD1	RF.RD2		A	В	ALU			
ORI	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[20:16]	RF.RD1		IR[15:0]	A	EXT	ALU			
LUI	PC	IR[15:0]	NPC.NPC	PC	IM	ALUOut	IR[20:16]	RF.RD1		IR[15:0]	A	EXT	ALU			
BEQ	PC	IR[15:0]	NPC.NPC	PC	IM			RF.RD1	RF.RD2		A	В	ALU			
JAL	PC	IR[15:0]	NPC.NPC	PC	IM	PC.PC4	0x1F									
												ना		INTANS	ም // ሆ	1,60

RF

NPC

提纲

- 数据通路一般性方法
- ▶ 状态/控制信号矩阵压缩表达
- 2个数据通路设计的对比

常规表达方式的不足

- S1: 信号/状态矩阵: 每条指令一张表
 - □ N条指令N张表
- S2: 信号真值表矩阵有效表达密度低
 - □ 大量0元素
 - □ 我们真正需要却是1元素

		Λ	Firmat	co	C1	ca	co	c/	1 6	- 4	ce	67	co	SS
	0	n Eur	CI	n c	1 0	2 C	2 6	л	CE	cد	67	co	59	
	Op	Funct	S0	S1	S2	S3	S4	S5	S6	S7	S	BS	9	Н
PCWr			1	0	0	0	0							
NPCOp			PC+4	Χ	X	Χ	Χ							Н
IRWr			1	0	0	0	0							
GPRWr			0	0	0	0	1							Н
DMWr			0	0	0	0	0							Н
ALUOp			X	add	add	add	add							H
GPRSel			X	00	00	00	00							
WDSel			X	01	01	01	01							
ExtOp			Х	SE	SE	SE	SE							
BSel			Х	1	1	1	1							_

	Op	Funct	S0	S1	S2	S 3	S4	S 5	S6	S7	S 8	S 9
LW	100011		1	0	0	0	0	0	0	0	0	0
SW	101011		1	0	0	0	0	0	0	0	0	0
ADDU	000000	100001	1	0	0	0	0	0	0	0	0	0
SUBU	000000	100011	1	0	0	0	0	0	0	0	0	0
ORI	001101		1	0	0	0	0	0	0	0	0	0
LUI	001111		1	0	0	0	0	0	0	0	0	0
BEQ	000100		1	0	0	0	0	0	0	0	0/1	0
JAL	000011		1	0	0	0	0	0	0	0	0	1

指令分类的启发

▶ 由于指令分为R、I、J等少数几类,并且指令

格式非常简洁, 因此:

□ 数据通路: 变化点很少

□ 控制逻辑

▪ 控制信号

固定

□ 固定的原

	Up-JAL
辑: 状态机可以按大类划	S2 S6 S8 S9 Jmp
: 大部分取值非常	S3 S5 MW
	S4 MemWB
周期,固定的值	

	N	NPC		NPC		NPC		NPC		NPC		nc	ID 4	10		RF	_		EVT	Α	LU	ALLIQUA	DM	DB
	PC	IMM	PC	IM	IR	WrData	WrReg	А	В	EXT	1	2	ALUOut	DM	DR									
指今						变	变					变												

指令分类的启发

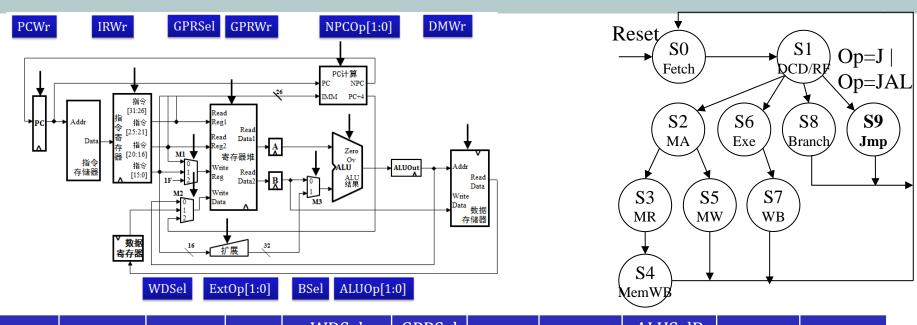
由于指令分为R、I、J等少数几类,并且指令 格式非常简洁,因此:

□ 数据通路: 变化点很少

□ 控制逻辑: 状态机可以按大类划分

	NPCOp	PCWr	IRWr	WDSel (MUX)	RegDst (MUX)	RegWr	ExtOp	ALUSelB (MUX)	ALUOp	MemWr
ADDU	S0:PC+4	S0	S0	S7:`ALUOut	S7:`RD	S7			S6:ADD	
SUBU	S0:PC+4	S0	S0	S7:`ALUOut	S7:`RD	S7			S6:SUB	
ORI	S0:PC+4	S0	S0	S7:`ALUOut	S7:`RT	S7	S6:`UEXT		S6:0R	
LW	S0:PC+4	S0	S0	S7:`DM	S4:`RT	S4	S2:`SEXT	S2:2	S2:ADD	
SW	S0:PC+4	S0	S0					S2:2	S2:ADD	S5
BEQ	S0:PC+4 S8:BEQ	S0 S8·Zero	S0						S8:SUB	
– JAL	S0:PC+4 S9:JMP	S0 S9	S0	S9:NPC	S9:`1F	S9		S1:3	S9:ADD	-

你发现了什么?



	NPCOp	PCWr	IRWr	WDSel (MUX)	GPRSel (MUX)	GPRWr	ExtOp	ALUSelB (MUX)	ALUOp	DMWr	
ADDU	S0:PC+4	S0	S0	S7:`ALUOut	S7:`RD	S7			S6:ADD		
SUBU	S0:PC+4	S0	S0	S7:`ALUOut	S7:`RD	S7			S6:SUB		
ORI											
LW											
sw ◆ 不再关心无效值											
BEQ	SU:PC+4 S8:BEQ	Su S8·Zero	S0						S8:SUB		

S9:`1F

S9

S9:ADD

S1:3

S0:PC+4

S9:JMP

JAL

S0

S9

S₀

S9:NPC

怎么写表达式?

PCWr = S0 + beq·S8·zero + jal·S9

PCWr = (lw+sw+addu+subu+ori+lui+beq+jal)·S0 +

beq·S8·zero +

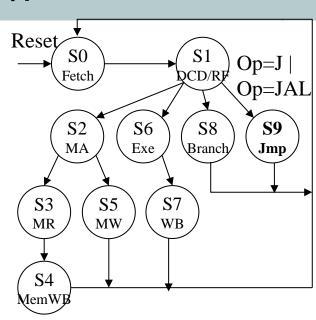
jal·s9

昨天的PPT

	NPCOp	PCWr	IRWr	WDSel (MUX)	RegDst (MUX)	RegWr	ExtOp	ALUSelB (MUX)	ALUOp	MemWr
ADDU	S0:PC+4	S0	S0	S7:`ALUOut	S7:`RD	S7			S6:ADD	
SUBU	S0:PC+4	S0	S0	S7:`ALUOut	S7:`RD	S7			S6:SUB	
ORI	S0:PC+4	S0	S0	S7:`ALUOut	S7:`RT	S7	S6:`UEXT		S6:0R	
LW	S0:PC+4	S0	S0	S7:`DM	S4:`RT	S4	S2:`SEXT	S2:2	S2:ADD	
SW	S0:PC+4	S0	S0					S2:2	S2:ADD	S5
BEQ	S0:PC+4 S8:BEQ	S0 S8·Zero	S0						S8:SUB	
- JAL	S0:PC+4 S9:JMP	S0 S9	S0	S9:NPC	S9:`1F	S9		S1:3	S9:ADD	7 VI 7 T'V

状态机的设计思路

- 指令划分为若干大类
- 状态机按设计大类
 - □ FETCH→DCD/RD是共性基础
 - □ 分支均在DCD/RD处判定



- 若某条指令归入任一分类都感觉不尽合理
 - □ 方法: 设置一个新的分支

按照指令大类设计的意义

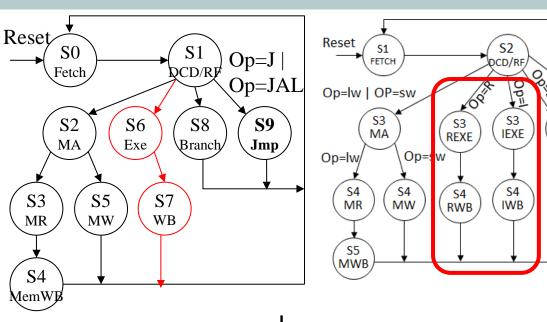
- Q: R类的主要区别是什么?
 - □ A: ALU应该执行什么操作!
- Q: 怎么解决?
 - □ A: 根据指令设置正确的ALUOp即可
- 从状态机的角度,不需要关心到R类内部细节
 - □ 只需要关注R类的执行策略和路径
 - 路径某个节点发生了什么,是低一个层次问题
 - ◆ 在指令/信号矩阵解决

有没有标准的状态机?

• A: 没有!

• 均可行

□ 差别很细微



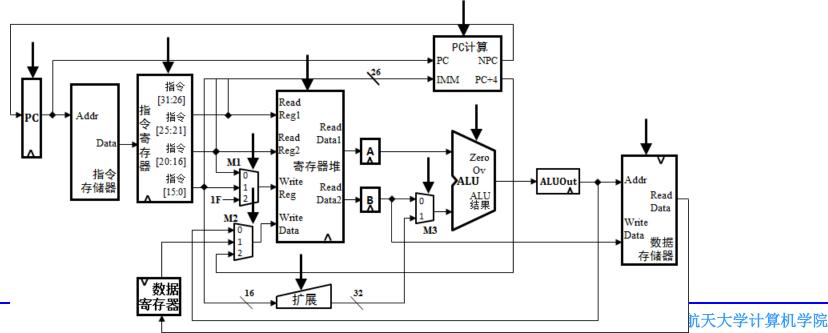
Op=J

S3

JMP

S3

BR

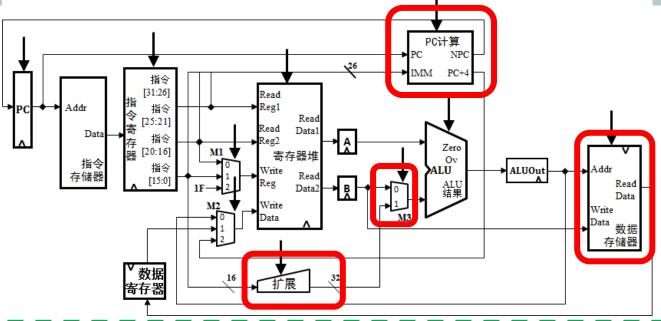


提纲

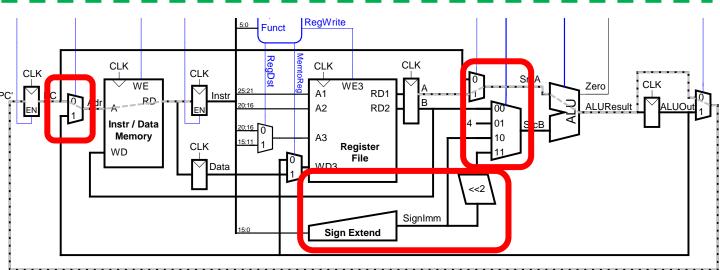
- 数据通路一般性方法
- ▶ 状态/控制信号矩阵压缩表达
- 2个数据通路设计的对比

PPT数据通路 vs 教科书数据通路

PPT 数据通路



教科书 数据通路



PPT数据通路 vs 教科书数据通路

- ▶ PPT: 保留NPC, 与单周期、流水线一致
 - □ 整个设计更加简洁,数据通路的结构化特征更好
 - ◆ 设计思想: 高内聚, 低耦合
 - 所有的与地址产生的都放在NPC中(包括今后的启动地址和异常地址)
 - □ IM地址端, ALU的A端: 没有MUX
 - ◆ 教科书: 为了J类指令
- ▶ 教科书:虽然可节省一点逻辑,但清晰度不好
 - □ 只有1个ALU, 节省了NPC里的加法器
 - □ 状态机设计更加复杂,更有利于学习"状态机"

谢谢大家!